Conditional Averaging a New Algorithm for Digital Filter

Sukesh Rao M, NMAMIT,
E&C Dept, Nitte, India.
Email: sukesh_muligar@yahoo.co.in
Dr. S. Narayana Iyer, NMAMIT, Nitte, India.
Email: iyer_narayana@yahoo.com

***Abstract*—This paper aims at designing a new algorithm for digital filters. The traditional methods like FIR, IIR have been improved in recent times with new approaches. However, the developments have used complex arithmetic calculation and dedicated DSP processors. In this research project, effort has been made to reduce such complexities using a procedure based on the technique of Conditional Averaging. The entire algorithm is developed using more of conditional statements and less of arithmetic calculations.**

Digital signals are filtered at different stages of signal processing. However high speed processor is used for different calculations associated with filtration process. An averaging is one such scheme used in simple FIR filter, which performs low pass filtering operation. Conditional Averaging is a new technique, which is one of the improvements in continuous time averaging. Conditional Averaging algorithm is explained in this practice with different examples for the design of low pass filter. This algorithm has been successfully tested using digital starter kit with TMS3206416v DSP processor. Using code composer studio, the entire algorithm is written in C/C++ language and compiled into an assembly language. Conditional averaging can be implemented with any general purpose processor to arrive at other types of filters with certain necessary modifications.

***Index Terms*— FFT, DSP, LPF**

I. Introduction

Filter is a system which selectively changes the wave shape, amplitude, frequency content and phase characteristic of a signal as desired. In digital signal processing, based on the design constraints digital filters of FIR, IIR or even other types are commonly in usage. In real time signal processing, these filters use dedicated Microprocessor system to carry out complex floating point arithmetic operation. To acquire high accuracy and high precision of the system response, high frequency supportive DSP processor is used. The scope of this work namely Conditional Averaging is to minimize the complex arithmetic operation and obtain better response for the required design using general purpose microprocessor.

A. Problem Definition

A specially devised technique of conditional averaging has been developed to design a low pass filter. The term 'conditional averaging' has been assigned to this method as it performs the averaging of a given data set based on certain attributed conditions. This algorithm has been tested for different cutoff frequencies maintaining the sampling frequency at a fixed rate. MATAB simulation technique has been used to confirm and validate the algorithm. The algorithm is verified with the following criteria.

1) For a fixed sampling frequency, the sample of the incoming signal is processed for three different levels of conditionality, resulting in three different types of cutoff frequencies or the signal bandwidth.
2) A relation is formed between the conditional statements and the cutoff frequencies which remain true for any number of remaining conditional statements and the desired cutoff frequencies.
3) A mixed harmonic signal is fed to the system to obtain the system response for different cutoff frequencies.

II. CONDITIONAL AVERAGING- THE PROPOSED TECHNIQUE

Conditional averaging is a new scheme proposed in the area of different averaging techniques [1]. Simple averaging with N points will reduce any of the AC components corresponding to the amplitude variations. All averaging techniques act as low pass filter with the filter coefficients equal to 1/N. conditional averaging will not use any such kind of filter coefficients and any definite mathematical relation as such.

The Fast Fourier Transform (FFT) is the method used in the DSP (digital signal processing) to find out the frequency spectrum of any given signal. It is a mathematical operation for obtaining the accurate frequency spectrum. It is also possible to predict the frequency spectrum of the signal with signal amplitude variations. The spectrum so obtained may not be accurate and it is approximated. If we know the sampling frequency or the timing interval of the incoming signal, it is possible to predict the harmonics present in the signal. Figure 1 shows a random signal. We can determine the frequency plot accurately for the signal shown in figure 1 using the frequency transform techniques. Also possible to predict approximately the harmonics present in the signal by knowing the sampling frequency. Let us assume that the signal is sampled at a rate of f_s Hz and figure 2 (a) shows the resulting discrete samples.

From the sampled signal, one can conclude that, (1/ f_s) is the time gap between two samples. If the signal varies with more number of successive samples, then it corresponds to lower harmonics
Fluctuation with lesser number of successive samples corresponds to higher harmonics. All such samples
which contribute to the frequency component less than fs can be easily predicted. These samples are marked and shown in figure 2(a). The samples marked in red, blue and yellow color; corresponds to different frequency components within the bandwidth fs. The higher harmonic

DOI: 01.IJSIP.01.03.188

is marked in red color and it contains approximately 7 samples. This corresponds to a frequency component (1/7)fs. The next harmonic is marked in blue color. It

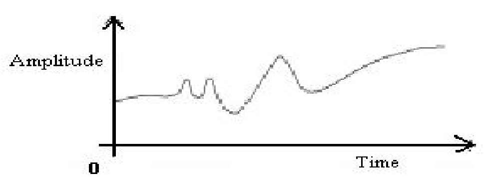
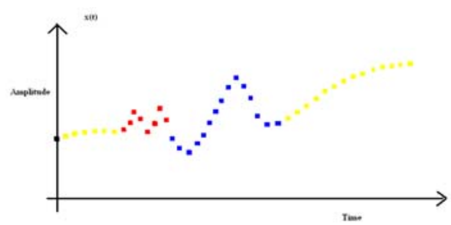


Figure 1. An arbitrary Signal

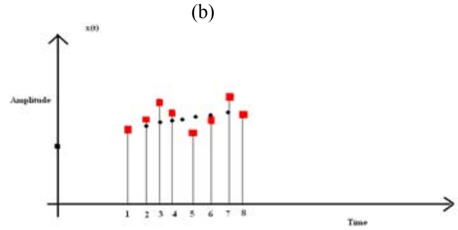
consists of approximately 15 samples, which contributes to (1/15)fs frequency spectrum of the signal. Similarly the yellow colored sample set belongs to lowest harmonic of the signal and this frequency is less than (1/40)fs.

To suppress the two harmonics with samples red and blue colors and maintain only the harmonic lower than

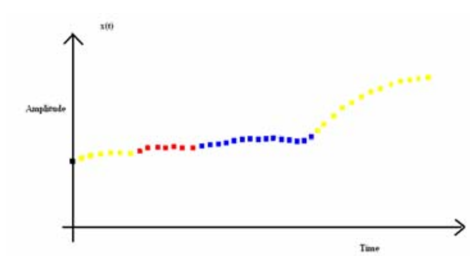


(a) Samples set identified with different color

(b)



(b) Magnified view of samples with red color



(c) Final reconstructed Signal

Figure 2

(1/40)fs, we can think of a technique which will process only these samples. Continuous averaging will suppress the higher frequency components depending on the number of samples considered for the averaging. Apart from this type of averaging one more kind can be introduced, which performs the sample modification similar to the averaging. Consider red colored samples from the figure 2 (b) for the sample modification. To remove the fluctuation of this signal, a linear path is predicted between first and the last sample. The linear path is formed between time index t=1 and t=8. It is shown in figure 2 (b). If there are even number of samples, then a value is obtained at the center [t=4.5] of first and last sample. From this value at t=4.5, the averaging with first and last sample leads to two more new sample sets corresponding to t=3 and t=6. Same method of averaging is performed to obtain the values at t= 2, 4, 5 and 7.

The newly obtained values will make the signal shown in figure 2 (c). Same kind of averaging is performed to the yellow colored samples. Since we perform the averaging only for a set of samples with certain conditions on amplitude variations, this algorithm is named as “conditional averaging”. It indicates that this proposed algorithm results in reduced computations, which is certainly an overwhelming advantage.

Signal averaging is performed on the basis of the variation in the magnitude of the incoming samples, the averaging takes place always between two samples. Averaging of two samples is performed over N points to determine the new possible sample of the signal. Not all the incoming signal sample undergo for the averaging operation.

To test this algorithm, consider a signal, which is sampled at a rate of f_s with four different frequency components f_1, f_2, f_3 and f_4 with different phase angle Φ1, Φ2, Φ3 and Φ4. We express this signal as follows.

$$y = a_1 \sin(2\pi f_1 t + \varphi_1) + a_2 \sin(2\pi f_2 + \varphi_2) + a_3 \sin(2\pi f_3 t + \varphi_3) + a_4 \sin(2\pi f_4 t + \varphi_4)$$

By assuming suitable values of these frequencies and phase angles, the simulation of the conditional averaging is carried out. Consider f1, f2, f3 and f4 to be 600 Hz, 50 Hz , 2000 Hz and 800 Hz respectively with phase angle of 2Π/3, 0, Π/5 and Π/2.

$$y = 10 \sin\left(2\pi 600t + \frac{2\pi}{3}\right) + 40 \sin(2\pi 50t) + 15 \sin\left(2\pi 2000t + \frac{\pi}{5}\right) + 18 \sin\left(2\pi 800t + \frac{\pi}{2}\right)$$

DOI: 01.IJSIP.01.03.188

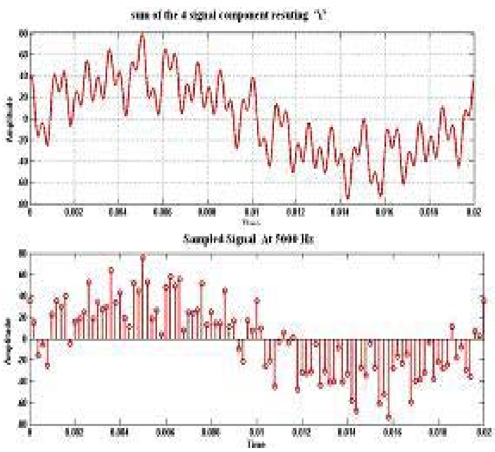


Figure 3 Resulting wave for 100 samples after sampling

Conditional averaging is performed for the first 100 samples of the sampled input signals as shown in figure 4 (a) to get different cutoff frequency for a LPF. The conditional averaging is carried out for three different cases
a) Four sample-conditional Averaging
b) Eight sample- conditional Averaging
c) Sixteen sample- conditional Averaging

A. Four sample-conditional Averaging

Four sample-conditional averaging uses only four sample buffers. The samples in the buffer are used for the comparison process to check the conditions of a relational set. A relational set is an array, which is obtained by comparing two samples each from the main sample set and contains combination of only two values. If the signal frequency of interest is f, then the sampling should be carried out at the rate more than 4f. It is also possible that if a signal is sampled at fs, then the signal can be band limited to minimum of fs/4. Conditional averaging with four sample is the minimum possible selection of the cutoff frequency for the design of LPF.

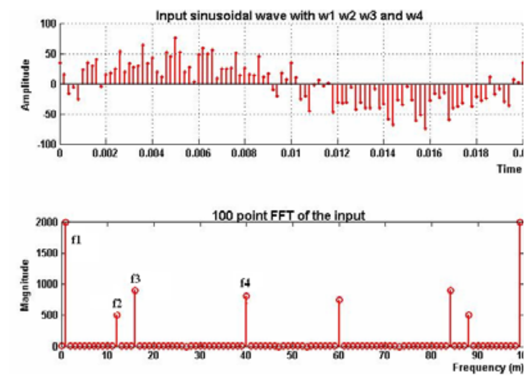
The Algorithm uses following steps
1) Take new signal sample to the buffer of size four in first-in last-out order.
2) Check the relation between all 4 samples by getting a new set of relational array with 3 elements.
3) In the relational array search for the unwanted sequence and eliminate that sequence in the buffer by performing Averaging of four samples.
4) From the updated buffer, the last sample is taken as output and entire process is repeated from step 1.

In the figure 3 or figure 4 (a), samples are taken at 5000Hz sampling rate and a 100 point FFT is obtained for the frequency analysis. Here m varies from 0 to 99 and m=100 correspond to the Nyquist rate, i.e. 5000 Hz in this case.
To find the frequency of analysis 'f' from the above FFT

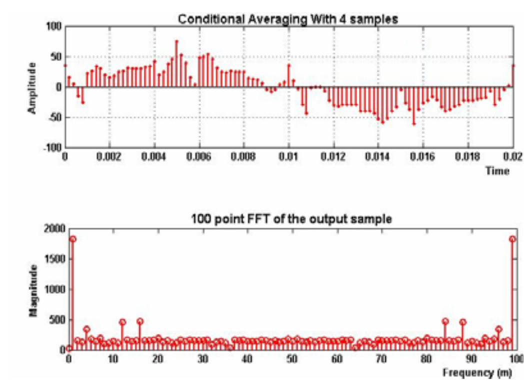
$$f = \frac{mfs}{n}$$

Where N=100 an fs =5000 Hz
The four samples conditional averaging will result in a bandwidth at $fs/_4$
i.e. $5000Hz/_4 = 1250Hz$
After performing the conditional averaging, the resulting sample is shown in figure 4 (b) with its FFT.

The frequency 1250 Hz corresponds to the m=25 and all the values with m>25 are attenuated. Hence it is



(a) Input FFT



(b) Output FFT

functioning like a LPF for the designed frequency fs/4. The process introduces a delay of 4 sample time between an input and the output value at any instant of time. The algorithm mentioned for the conditional averaging is implemented in the C language.

B. Eight sample-conditional Averaging

Four sample-conditional averaging is simpler and the basic averaging. Whereas *eight* sample-conditional averaging includes the signal constraints of 4 sample conditional averaging also. An 8 sample buffer is maintained for the comparison process to check for the conditions. Eight sample conditional averaging gives a signal bandwidth of fs/8. Conditional averaging with eight sample uses the algorithm used similar to that of *four* sample-conditional averaging to get the bandwidth for the desired of LPF.

To find the frequency of analysis 'f' from the FFT shown in figure 5.

$$f = \frac{mfs}{N}$$

Where N=100 an fs =5000 Hz
The *eight* sample-Conditional Averaging will give cutoff value at fs/8. i.e. 5000/8=625 Hz .This can be verified by figure5**.**

The frequency 625 Hz corresponds to m=12.5 and all the values m>12.5 are attenuated. Hence it is acting like a LPF for the desired bandwidth. The whole process introduces a delay of 8 sample time between an input and the output.

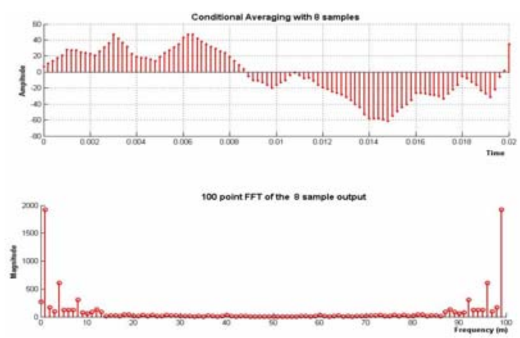


Figure 5. Output FFT of 8 sample conditional averaging

C. Sixteen sample -conditional Averaging

In this Averaging technique only *sixteen* sample buffers are maintained for the comparison process to check the conditions. *Sixteen* sample-conditional averaging includes the both the processing stages of four and eight sample averaging. For a sampling frequency fs, the *sixteen* sample conditional averaging given a bandwidth limit of fs /16 in the design of LPF. To find the frequency of analysis 'f' from the above FFT shown in figure 6.

$$f = \frac{mfs}{N}$$

where N=100 an fs =5000 Hz. The *eight* sample-conditional Averaging will give cutoff value at fs /8. i.e. 5000/16=312.5 Hz .This can be verified by figure 6.

The frequency 312.5 Hz corresponds to m=6.25 and all the values m>6.25 are attenuated. Hence it is behaving like a LPF. The whole process introduces a delay of 16 sample time between an input and the output.

FIR filtering obeys sinusoidal functional manipulation, hence the output samples are also the part of sine or cosine functions. The above work is still modified for the better performance for different frequencies. When more samples are taken for a particular conditionality, it is possible to modify the range of the bandwidth. Conditional Averaging makes use of the deterministic signals like ECG, EMG and EEG, which are the better sources of signals for the analysis. To reiterate, this algorithm reduces the computation.

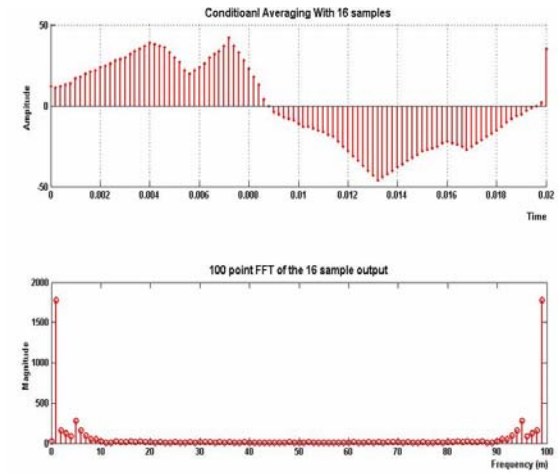


Figure 6 outputs FFT of 16 sample conditional averaging

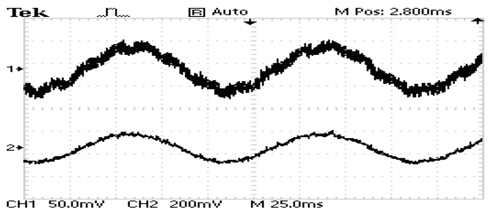
III. RESULT

Conditional averaging algorithm is verified in real time to compare the result of the theoretical simulation. To perform this operation DSP Starter kit (DSK6416) is used for the programming and the debugging. TMS320C6416v is the key element in DSK6416 system and it has got all the other speech processing peripherals interfaced to it. Input signal are fed using audio codec and then processed by the DSP processor.

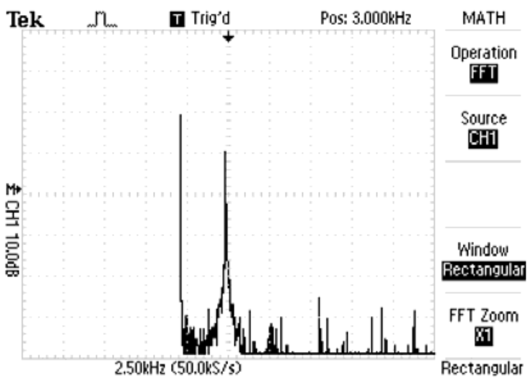
Audio codec performs analog to digital conversion and vice versa. Analog signals are mixed using simple passive components to get the mixture of all the input signals. Signal selections are made such that, they will be within the sampling frequency of the system. The sampling frequency of the DSK6416 is selected by the program command to set at 8 KHz. All input signals are band limited to less than 4 KHz to satisfy the Nyquist rate.
A real time testing has been carried out for sixteen sample conditional averaging. *Sixteen* sample-conditional averaging gives a very small bandwidth of fs /16. More conditional statements has to be included in the processing of the signal, given that the number of samples considered are more for the signal processing. The time domain input and the output signal after the processing through conditional averaging are shown in the figure 7 (a).

A test is conducted with different input FFT, wherein a signal of higher frequency component within the input band limit is fed into the system. The output FFT obtained clearly shows the attenuation of the signal above 500Hz in the spectrum. The signals above 4 KHz are also attenuated, as the signal was sampled at 8 KHz rate and hence it can not reconstruct the signal beyond 4 KHz (refer figure 7 (c)). Including more conditional statements for a signal of higher frequency value, the output response of the system we find is much better.

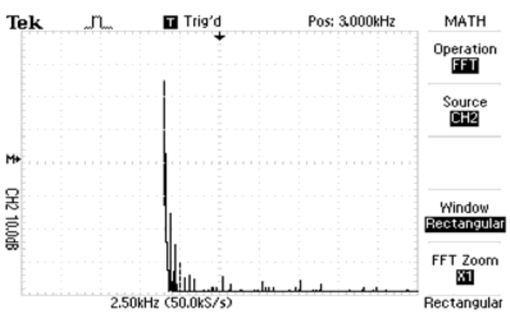
DOI: 01.IJSIP.01.03.188



(a) Input and Output signal



(b)Input signal FFT



(c) Out put FFT

Fig 14

CONCLUSION

The desired bandwidth for the proposed low pass filter could be achieved after the simulation and debugging of the algorithm using the conditional averaging, which works as a low pass filter. It may be noted that filter uses simple two values averaging at a time and gives the corrected output by consuming less processing time. Normally time domain averaging uses all the incoming samples for the processing, but the conditional averaging uses only limited number of samples, which satisfy the conditional array set. This results in reduced computational time and burden on the processor.

Any type of general purpose processor can be used to design the system for the signal processing using this algorithm. Thus, unlike other algorithm, this method requires only a general purpose processor. However, in the proposed algorithm, in order to verify all the sets of possible combination within the short period of time, before the next sample arrives, a high speed general purpose processor is required. This processor should be able to execute minimum of 100 to 1000 instructions within one sampling time. For the lower bandwidth, system undergoes more conditional statements and therefore it requires more instruction for the processing.

Conditional algorithm can also be developed for the analysis of ECG, EEG signals as well. Therefore, conditional averaging becomes a very useful filtering algorithm to analyze the biomedical signals. All such biomedical signals are of low frequency and any high frequency noise or the signal surge can be easily detected using the conditional averaging.

The main drawback of the system is that, it may distort the signal by adding many other lower harmonics within the bandwidth. To alleviate this and get a desired bandwidth, it is necessary to include more conditional statements during processing.

This conditional averaging is possible to be applied to high pass filtering operation also. It may not be just averaging as performed in the case of the low pass filter design. Design of narrow band pass filter using the conditional averaging algorithm is another future work.

REFERENCES

[1] Welch P. D " The use of Fast Fourier Transform for the estimation of power spectra: A method based on time averaging over short, Modified periodgrams" *IEEE transaction on audio and electroacoust.*, Vol. AU-15 No.2 June 1967

[2] Richard G Lyons, Understanding Digital signal processing Pearson Education third edition.2001.

[3] TMS320C6713 DSK Technical Reference, SPECTRUM DIGITAL, INC

[4] TMS320 DSP datasheet by Texas instrumentation.

[5] Emmanuel C. Ifeachor, Digital Signal Processing, Pearson Education

DOI: 01.IJSIP.01.03.188